ARMY RESEARCH LABORATORY

A Means for Incorporating Time-Dependent Phenomena in Existing Vulnerability Analysis Methods

Phillip J. Hanes

ARL-TR-738

April 1995

NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute endorsement of any commercial product.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1995	Final, July 1993–December 1993

4. TITLE AND SUBTITLE

A Means for Incorporating Time-Dependent Phenomena in Existing Vulnerability Analysis Methods

5. FUNDING NUMBERS

PR: 1L162618AH80

6. AUTHOR(S)

Phillip J. Hanes

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: AMSRL-SL-BA
Aberdeen Proving Ground, MD 21005-5068

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-738

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

There are many phenomena addressed by the vulnerability analysis community which are, in truth, time dependent. However, due to computational constraints, whether actual or historical, most such phenomena are treated in a manner that ignores or, at best, crudely approximates this time dependency. This report describes a method which could allow such dependencies to be added to existing vulnerability analysis software in a more physically realistic manner. It also describes a possible implementation of these ideas within a vulnerability analysis code.

14. SUBJECT TERMS

event queues, vulnerability, time-dependent phenomena

15. NUMBER OF PAGES

19

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

		Page
1.	THE NEED FOR TIME DEPENDENCY	1
2.	CURRENT RAY TRACING PARADIGM (GEOMETRY DRIVEN)	1
3.	TIME ORDERED RAY SEQUENCING (EVENT DRIVEN)	3
4.	A SAMPLE APPLICATION (MUVES)	5
5.	CONCLUSION	7
6.	REFERENCES	8
	APPENDIX A: TIME SEQUENCE INTERFACE DEFINITION	9
	DISTRIBUTION LIST	11

INTENTIONALLY LEFT BLANK.

1. The Need for Time Dependency

There are numerous time dependent phenomena which must be addressed in the process of performing a vulnerability analysis on a given system. Many of these involve events that occur after the initial impact of the threat (e.g., leakage from fuel or hydraulic lines). Such post-impact events have been incorporated into some existing vulnerability methodologies.

There is, however, a class of time dependent effects which does not seem to have been incorporated into any vulnerability methodology in a physically realistic way. These effects take place in the microseconds immediately following a threat impact or detonation. They happen so quickly that they are usually treated simultaneously with the threat impact.

These phenomena include such things as buckling plates, aerosolization of fuel, punching holes in components, electric arcing, and so on. Since the sequence of most analyses is driven primarily by the geometry of the target, these effects are difficult to consider using current software. Therefore, the usual approach is to ignore such effects if they occur outside the normal flow of the geometry. For example, when determining whether a fire has started, most programs look for sparks along the ray trace which perforates a fuel cell but ignore sparks or incendiaries along nearby ray traces. Similarly, holes produced by one fragment are virtually always ignored when considering fragments coming later in time. Such short cuts may be expedient, but they ignore important information which could affect the results of the analysis.

It should be possible to incorporate such effects so that physical considerations (geometry, perforation, etc.) are driven by the order in which events occur. Doing so would allow a more physically realistic examination of synergistic effects.

2. Current Ray Tracing Paradigm (Geometry Driven)

Many vulnerability analyses performed today use ray tracing as their primary means of geometry interrogation. There are more sophisticated methods under development which could augment the use of rays, but most of these are not yet available. Furthermore, it is not likely that they will completely replace rays for vulnerability requirements, since many phenomena travel in straight lines when interacting with a target.

Therefore, since rays are used extensively and will probably continue to be highly useful in the foreseeable future, it seems necessary to continue improvements to software which depends on this capability. This reportdescribes the current use of rays for vulnerability analyses and then develops a method for time dependent ordering in the context of this description.

To the extent possible, implementation details for any particular ray tracing package or vulnerability analysis software are avoided in order to develop a more generic description of the capabilities envisioned. An application of these ideas to the Modular UNIX-based Vulnerability Estimation Suite (MUVES) [1] follows at the end of this report.

For those unfamiliar with ray tracing, a ray trace is a linear path through a geometric model (often called a "target description" in vulnerability analyses). Inside the computer, the geometric information for a ray trace is stored as a series of hit points, segment lengths, normal vectors, curvatures, and perhaps other values, depending on the data required by the analysis. The primary data structures contained in a ray trace might be visualized as shown in Figure 1, where each box represents a component hit by a ray trace through a target and the lines indicate the sequence in which they were hit.

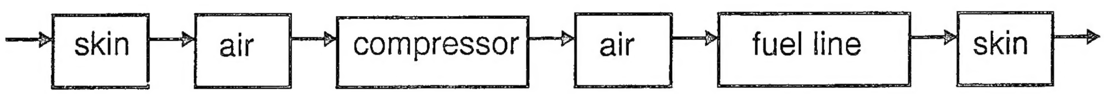

Figure 1. Ray Trace representation

In a vulnerability analysis, some energetic phenomenon (called a *threat*), whose propagation mechanics are governed by a set of equations (called *penetration equations* for armor-piercing munitions or fragments), travels along this path, interacting with the components encountered on the path, losing energy, and usually causing damage to the components. The process of computing the degradation and damage can be called *analyzing* a ray trace.

These ray trace segments are usually analyzed sequentially; that is, one ray trace is analyzed from beginning to end (or until the threat no longer has the capability

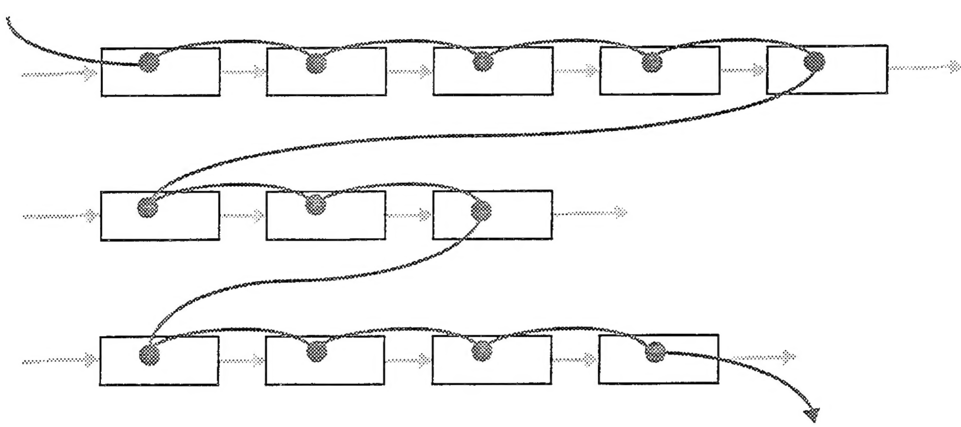

Figure 2. Normal sequence of ray trace analysis for three rays

to damage the target), and then the next ray trace is analyzed. This sequence is repeated until the entire collection of ray traces is fully analyzed. Using the previous representation of a ray trace, the sequence of analysis might be illustrated as shown in Figure 2.

Some modern vulnerability software supports ray tracing on demand rather than computing the collection of all possible rays before any analysis begins. While this is a significant improvement in both efficiency and realism, allowing the ray selection to be determined by computations during the course of the analysis, it does not necessarily introduce a time ordered approach to analyzing the ray traces.

3. Time Ordered Ray Sequencing (Event Driven)

In order to use these rays in some time dependent fashion, it is necessary to change the order in which they are processed. Conceptually, this is a fairly simple task; threat/component interactions are analyzed in the order in which they would happen in a real event. However, cleanly incorporating such a sequence in a program requires some finesse as well as a clear understanding of both the computer science issues and the vulnerability issues involved.

Figure 3 shows a possible sequence for a time ordered analysis of the same set of three ray traces introduced in Figure 2. In this figure, the dark line shows the time ordered sequence of analysis, and the circles represent *events* where the threat traveling along the ray trace impacts the component represented by the box containing the circle. Each event occurs at a particular time, *T*, some examples of which are shown in Figure 3. In this fashion, the software switches from one threat to the other while determining the effects of all the threats initiated during the analysis.

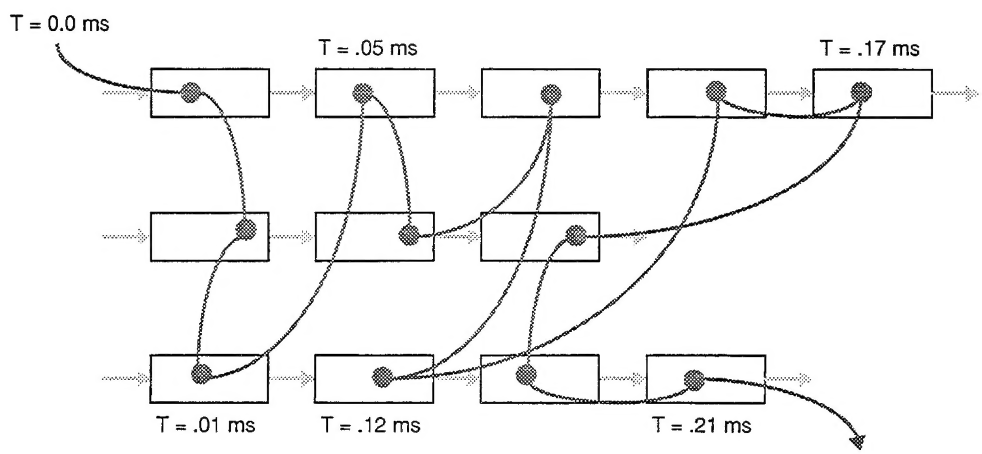

Figure 3: Time ordered sequence of ray trace analysis

Comparing Figure 3 with Figure 2 gives some insight concerning why this has not generally been implemented in vulnerability software. The sequence of events is confusing even for a small number of threats; for a large number of ray traces (a minimum of several hundred for a typical fragmenting threat), the complexity would scale up quickly. The software, in addition to keeping track of where it is in the collection of ray traces, must also keep track of *every* threat propagating along every ray trace at any given moment, since *a priori* there is no certainty which interaction will be performed next. This requirement alone was prohibitive for many years, since computer memory was not large enough to hold the required volume of information. Our goal is to perform these evaluations while keeping the complexity to a manageable level.

In order to implement this concept in a vulnerability code, it is necessary to introduce an abstraction called an *event queue*. An event queue is a list of events which are sorted in the order in which they occur. Thus, the first event is the first item encountered when traversing the list. When a new event is placed on the queue, it is inserted in its time dependent order--that is, after prior events but before later events. The interval between the events is not relevant, only the order in which they occur. As events are processed, they are removed from the queue, and new events may be placed on the queue as a result of other events. Thus, the event queue is a dynamic list which, at any time during the analysis, contains the set of events that are known to be pending.

Therefore, by thinking of the impact of a fragment on a component as an event, the spaghetti in Figure 3 can be sorted into a linear list of known events, much like the original simple concept mentioned at the beginning of this section. This type of event queue is shown in Figure 4 with a hypothetical operation on the first event. Note that this generates another event, which is then inserted onto the queue.

To implement these ideas in a vulnerability code, there are a number things which must be done in succession.

1. Trace all rays required to represent the effects of a given threat (e.g. a high-explosive munition). Note, this may include rays representing more than one damage mechanism (e.g. fragments and blast wave).
2. For each ray trace, compute the time of the *first* component impact (only) along that ray trace and insert the threat parameters and ray trace segment onto the event queue.
3. Pop the first threat/component interaction from the event queue and analyze the effects of that interaction.
4. As each event is analyzed, determine the time of the next impact *for that threat* (which will usually be along the same ray trace) and insert it onto the event queue. If the threat has run out of energy, discard the remainder of the ray trace. If the threat spawns one or more subsequent threats, perform steps 1 and 2 for the new threats.

5. Proceed to the next event on the queue and analyze it as in step 4. Repeat until there are no more events on the queue.

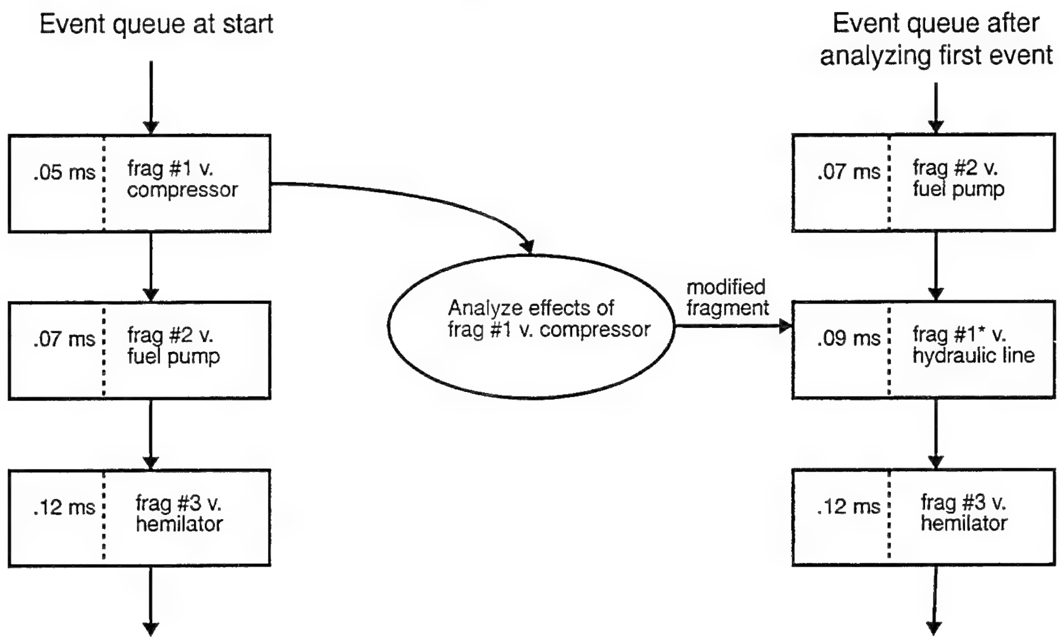

Figure 4. Event queue and results of analyzing the first event

By using data structures which contain both the component information and the threat information for a single event, the management headaches described earlier can be minimized, and the coordination of threat effects becomes virtually automatic.

It should be noted, however, that memory requirements will not be reduced by this method. A fairly large main memory will be required to perform these computations for anything other than trivial targets. Nevertheless, the resulting increase in physical realism should be worth the effort.

4. A Sample Application (MUVES)

In order to show how these ideas might be implemented in vulnerability software, this section develops minispecifications for a hypothetical burst module that might appear in the MUVES environment. The following example could be implemented using functions from the MUVES library and from the time sequence package defined in Appendix A.

This module assumes that some munition, perhaps a high explosive round, has burst. The decision regarding whether the round bursts is assumed to have been made elsewhere.

Burst module specification :

establish time of burst

determine number of fragments

for each fragment required to simulate burst

    determine parameters of fragment (must include direction vector)
    ray trace path of fragment
    determine time of first impact (from velocity and distance to component)
    attach threat parameters to component ray trace segment
    insert component onto event queue

while there are events in the event queue

    pop next event from the event queue
    determine appropriate module for threat/component interaction
    invoke interaction module

return (analysis is complete)

This is relatively straightforward; the principal requirement is the ability to determine the speed and direction of each fragment. Given the speed and direction, it is possible to determine the time of its impact and insert the interaction event onto the queue.

The next problem is to examine the interaction of a fragment with a component.

Fragment Interaction module specification :

perform perforation calculations

determine damage to component (if any) and store for later evaluation

if the fragment perforates

    determine parameters of degraded fragment
    * could deflect fragment at this point, if appropriate *
    assess immediate effects of perforation (holes, etc.)
    determine time of next impact
    attach threat parameters to component ray trace segment
    insert component onto event queue

else

    * could ricochet fragment at this point, if appropriate *

if the fragment spawns another threat (break up, for instance)

    determine parameters of new threat(s) (including direction vector)
    ray trace path of new threat(s)
    determine time of first impact
    attach threat parameters to component ray trace segment
    insert component onto event queue

return (burst module will select next event for analysis)

Although this paper uses a fragmenting effect to illustrate the use of a time sequence capability, this method could easily be extended to other threat phe-

nomena, such as a blast wave. The principal requirement is the development of interaction modules to perform computations for each type of threat of interest. Note that the effects of all phenomena would be interleaved through the use of the event queue. This is closer to the real phenomenology and is thus a highly desirable feature of this methodology.

5. Conclusion

It seems possible to add time dependent phenomena to an existing vulnerability analysis code without radically changing its structure.

If the original software has been developed in a modular fashion, this capability could be added by implementing event queue modules and some trivial functions to determine time to impact for various threats. The remainder of the necessary operations should already exist, since they are all needed for current vulnerability analysis software.

Adding such a capability could significantly increase the capability of existing software to analyze the synergistic effects of threats on military systems.

6. References

[1] Phillip J. Hanes, Scott L. Henry, Gary S. Moss, Karen R. Murray, and Wendy A. Winner. "Modular UNIX-based Vulnerability Estimation Suite (MUVES) Analyst's Guide." U.S. Army Ballistic Research Laboratory, BRL-MR-3954, Aberdeen Proving Ground, MD, December 1991.

Appendix: Time Sequence Interface Definition

What follows is an interface definition for a software library that provides the event queue capability described earlier in this document. This interface definition is written in the style used within the MUVES library.

<Ts.h> – MUVES "Ts" (time sequence) package definitions

The "Ts" package allocates multiple time sequence queues (event queues) to be handled within MUVES. Generic pointers are inserted into an event queue in time dependent order. The content of the pointers is to be determined by the calling function; Ts neither uses nor examines these pointers. Its sole purpose is to return them to the caller in time sequence order.

typedef TsHandle

The TsHandle is a reference to a time sequence queue. This handle is required when inserting new items onto the queue and when requesting items from the queue. This allows calling functions to maintain distinct queues simultaneously. These may then be used for different purposes if necessary.

TsHandle TsInitQueue()

TsInitQueue() initializes the internal structures for a time sequence and returns a handle to that queue.

bool TsRemQueue(TsHandle queue)

TsRemQueue() clears the memory associated with a given event queue. It should be called whenever the calling function is finished with the event queue. Only empty queues will be accepted by TsRemQueue(); if there is anything in the queue, TsRemQueue() returns false; otherwise the memory is cleared and TsRemQueue() returns true.

void TsInsertEvent(TsHandle queue, double time, pointer data)

TsInsertEvent() inserts a generic data pointer into the time sequence referred to by "queue". The value "time" is used to determine the correct place in the sequence for insertion. Values are inserted in ascending order. The value "data" is a generic data pointer which may refer to any piece of data required by the calling function. The Ts package makes no assumptions about the contents of the pointer nor attempts to de-reference that pointer.

pointer TsGetNextEvent(TsHandle queue)

TsGetNextEvent() determines the next event (lowest time value) in the time sequence referred to by "queue" and returns the pointer associated with that time. The queue entry for that pointer is removed.

void TsRemEvent(TsHandle queue, pointer data)

TsRemEvent() removes an event from the time sequence referred to by “queue”. The value “data” is a generic data pointer which matches one of the data pointers already inserted onto the queue. If it does not match any existing data pointer, TsRemEvent() silently ignores the request.

NO. OF COPIES	ORGANIZATION
2	ADMINISTRATOR ATTN DTIC DDA DEFENSE TECHNICAL INFO CTR CAMERON STATION ALEXANDRIA VA 22304-6145
1	DIRECTOR ATTN AMSRL OP SD TA US ARMY RESEARCH LAB 2800 POWDER MILL RD ADELPHI MD 20783-1145
3	DIRECTOR ATTN AMSRL OP SD TL US ARMY RESEARCH LAB 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DIRECTOR ATTN AMSRL OP SD TP US ARMY RESEARCH LAB 2800 POWDER MILL RD ADELPHI MD 20783-1145
	ABERDEEN PROVING GROUND
5	DIR USARL ATTN AMSRL OP AP L (305)

NO. OF COPIES	ORGANIZATION
1	OSD OUSD AT STRT TAC SYS ATTN DR SCHNEITER 3090 DEFNS PENTAGON RM 3E130 WASHINGTON DC 20301-3090
1	ODDRE AT ACQUISITION AND TECH ATTN DR GONTAREK 3080 DEFENSE PENTAGON WASHINGTON DC 20310-3080
1	ASST SECY ARMY RESEARCH DEVELOPMENT ACQUISITION ATTN SARD ZD RM 2E673 103 ARMY PENTAGON WASHINGTON DC 20310-0103
1	ASST SECY ARMY RESEARCH DEVELOPMENT ACQUISITION ATTN SARD ZP RM 2E661 103 ARMY PENTAGON WASHINGTON DC 20310-0103
1	ASST SECY ARMY RESEARCH DEVELOPMENT ACQUISITION ATTN SARD ZS RM 3E448 103 ARMY PENTAGON WASHINGTON DC 20310-0103
1	ASST SECY ARMY RESEARCH DEVELOPMENT ACQUISITION ATTN SARD ZT RM 3E374 103 ARMY PENTAGON WASHINGTON DC 20310-0103
1	UNDER SEC OF THE ARMY ATTN DUSA OR RM 2E660 102 ARMY PENTAGON WASHINGTON DC 20310-0102
1	ASST DEP CHIEF OF STAFF OPERATIONS AND PLANS ATTN DAMO FDZ RM 3A522 460 ARMY PENTAGON WASHINGTON DC 20310-0460
1	DEPUTY CHIEF OF STAFF OPERATIONS AND PLANS ATTN DAMO SW RM 3C630 400 ARMY PENTAGON WASHINGTON DC 20310-0400

NO. OF COPIES	ORGANIZATION
1	ARMY RESEARCH LABORATORY ATTN AMSRL ST DR FRASIER 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	ARMY RESEARCH LABORATORY ATTN AMSRL SL DR WADE WSMR NM 88002-5513
1	ARMY RESEARCH LABORATORY ATTN AMSRL SL E MR MARES WSMR NM 88002-5513
1	ARMY TRADOC ANL CTR ATTN ATRC W MR KEINTZ WSMR NM 88002-5502
1	ARMY TRNG & DOCTRINE CMND ATTN ATCD B FT MONROE VA 23651
	ABERDEEN PROVING GROUND
1	CDR USATECOM ATTN: AMSTE-TA
2	CDR USAMSAA ATTN: AMXSY-ST AMXSY-D
4	DIR USARL ATTN: AMSRL-SL, J WADE (433) AMSRL-SL-I, M STARKS (433) AMSRL-SL-C, W HUGHES (E3331) AMSRL-SL-B, P DEITZ (328)

NO. OF COPIES	ORGANIZATION
1	HQDA ATTN COL B B FERGUSON OUSDA A DDRE T&E THE PENTAGON WASHINGTON DC 20301-3110
2	HQDA ATTN R L MENZ OUSDA A DDRE R&AT ET THE PENTAGON RM 3D1089 WASHINGTON DC 20301-3080
1	HQDA ATTN LTG CIANCIOLO OFC OF THE ASST SEC OF ARMY RD&A WASHINGTON DC 20310-0100
1	HQDA ATTN DEP FOR SYS MANAGEMENT MG BELSTON OFC OF THE ASST SEC OF ARMY RD&A WASHINGTON DC 20310-0103
1	HQDA ATTN SAUS OR HON W HOLLIS THE PENTAGON RM 2E660 WASHINGTON DC 20310-0102
1	OUSD A DDT&E ATTN J O'BRYON THE PENTAGON RM 3D1084 WASHINGTON DC 20301-3110
2	OUSD A DDT&E ATTN A RAINIS THE PENTAGON RM 3E1081 WASHINGTON DC 20301-3110
1	OUSDA A DDT&E LFT ATTN J BLOOM THE PENTAGON RM 3E1084 WASHINGTON DC 20301
1	SAF AQT ATTN MR G WARREN THE PENTAGON RM BE939 WASHINGTON DC 20330-1000
1	ADV RES PROJ AGENCY ATTN MR B BANDY 3701 N FAIRFAX DR ARLINGTON VA 22203-1714

NO. OF COPIES	ORGANIZATION
1	DEPT OF THE NAVY ATTN RADM C R MCGRAIL JR THE PENTAGON RM 4E536 WASHINGTON DC 20350-2000
1	CDR ATTN AMCDE PI D MARKS US ARMY MATERIEL COMMAND 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	HQ ATTN AMCSCI R CHAIT US ARMY MATERIEL COMMAND 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	CDR ATTN AMCPD D GRIFFIN US ARMY MATERIEL COMMAND 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	CMDT ATTN AMXMC LS S US ARMY LOG MGMT COLLEGE FORT LEE VA 23801
1	DIR ATTN COMBAT DEVELOPMENT COL E TONEY US ARMY TRANSPORTATION SCHOOL FT EUSTIS VA 23604
2	DIR ATTN SLCRO MA DR J CHANDRA US ARMY RESEARCH OFC PO BOX 12211 RSRCH TRI PK NC 27709-2211
1	CDR ATTN SMCAR TD J KILLEN USA RDE CENTER PCTNNY ARSNL NJ 07806-5000
1	CDR ATTN STRBE FC ASH PATIL BELVOIR RDE CENTER FT BELVOIR VA 22060-5606

NO. OF COPIES	ORGANIZATION
1	CDR ATTN STRBE JDA M GOSS BELVOIR RDE CENTER FT BELVOIR VA 22060-5606
1	CDR ATTN ASQNC ELC IS L R USACECOM R&D TECHNICAL LIBRARY FT MONMOUTH NJ 07703-5000
1	CDR ATTN AMSMI RD J BRADAS US ARMY MISSILE COMMAND REDSTONE ARSNL AL 35898-5000
1	CDR ATTN AMSMI YTSD G ALLISON US ARMY MISSILE COMMAND REDSTONE ARSNL AL 35898-5070
1	CDR ATTN AMSMI REX W PITTMAN US ARMY MISSILE COMMAND REDSTONE ARSNL AL 35898-5500
1	DIR ATTN AIMS YLD V L STALLCUP US ARMY MISSILE AND SPACE INT CTR REDSTONE ARSNL AL 35898-5500
1	CDR ATTN AMCPM BLK III COL D DERRAH US ARMY TANK AUTOMOTIVE CMD WARREN MI 48397-5000
1	CDR ATTN AMSTA CK M ERICKSON US ARMY TANK AUTOMOTIVE CMD WARREN MI 48090

NO. OF COPIES	ORGANIZATION
1	CDR ATTN AMSTA CR MR WHEELOCK US ARMY TANK AUTOMOTIVE CMD WARREN MI 48397-5000
1	CDR ATTN AMSTA CV COL BECKING US ARMY TANK AUTOMOTIVE CMD WARREN MI 48397-5000
2	CDR ATTN AMSTA NKS D CYAYE J ROWE US ARMY TANK AUTOMOTIVE CMD WARREN MI 48397-5000
2	CDR ATTN AMSTA RG R MUNT R MCCLELLAND US ARMY TANK AUTOMOTIVE CMD WARREN MI 48397-5000
2	CDR ATTN AMSTA RSC J BENNETT W MICK US ARMY TANK AUTOMOTIVE CMD WARREN MI 48397-5000
1	CDR ATTN AMSTA RSK S GOODMAN US ARMY TANK AUTOMOTIVE CMD WARREN MI 48090-5000
1	CDR ATT AMSTA RY R BECK US ARMY TANK AUTOMOTIVE CMD WARREN MI 48397-5000
2	CDR ATTN AMSTA ZE R ASOKLIS AMSTA ZS D REES US ARMY TANK AUTOMOTIVE CMD WARREN MI 48397-5000

NO. OF COPIES	ORGANIZATION
1	OFC OF THE PEO ATTN SFAE ASM CV B BONKOSKY ARMORED SYS MOD WARREN MI 48397-5000
1	CDR ATTN ASST DEP CHIEF OF STAFF FOR COMBAT OPERATIONS HQ TRADOC FT MONROE VA 23651-5000
1	CDR ATTN ATAN AP M W MURRAY TRADOC FT MONROE VA 23651-5143
1	CDR ATTN MG STEPHENSON USA OPER TEST AND EVAL AGENCY 4501 FORD AVE ALEXANDRIA VA 22302-1458
1	CDR ATTN LTC G CRUPPER USA OPER TEST AND EVAL AGENCY 4501 FORD AVE 870 ALEXANDRIA VA 22302-1435
1	DIR ATTN ATRC WEC P SHUGART TRAC WSMR WSMR NM 88002-5502
2	DIR ATTN AMXIB MT AMXIB PS S MCGLONE US ARMY IND BASE ENG ACTIVITY ROCK ISLAND IL 61299-7260
1	CDR ATTN ATZL CAP LTC MORRISON DIR SURV TASK FORCE COMBINED ARMS COMBAT DEVELOPMENT FT LEAVENWORTH KS 66027-5300

NO. OF COPIES	ORGANIZATION
1	CDR ATTN ATZL HFM D SKELTON COMBINED ARMS COMBAT DEVELOPMENT FT LEAVENWORTH KS 66027-5300
1	DIR ATTN ATRC US ARMY TRADOC ANALYSIS CMD WSMR NM 88002-5502
1	DIR ATTN ATRC LS MR GREENHILL US ARMY TRADOC ANALYSIS CMD FT LEE VA 23801-6140
1	CDR ATTN ATZL CAI S USA COMB ARMS COMBAT DEV ACTIVITY FT LEAVENWORTH KS 66027-5300
1	CMDT ATTN ATSK CD USA ORD MISSILE & MUN CTR & SCHOOL REDSTONE ARSNL AL 35897-6500
1	CDR ATTN AMCAM USA MATERIEL COMMAND 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
2	CDR ATTN SMCAR-TDC USA ARDEC PCTNNY ARSNL NJ 07805-5000
1	DIR ATTN SMCAR CCB TL BENET WEAPONS LAB USA ARDEC WATERVLIET NY 12189-4050
1	DIR ATTN AMSAT R NR MS 2191 USA ATCOM AMES RESEARCH CENTER MOFFETT FIELD CA 94035-1000

NO. OF COPIES	ORGANIZATION
1	CDR ATTN AMSMI RD CS R DOC USA MISSILE COMMAND REDSTONE ARSNL AL 35898-5010
1	DIR ATTN ATRC WSR USA TRADOC ANALYSIS COMMAND WSMR NM 88002-5502
1	CDR ATTN CODE 1740 2 MR F FISCH CARDEROCK DIVISION NAVAL SURFACE WARFARE CTR BETHESDA MD 20084-5000
1	CDR ATTN 5164J J JOLLEY JTCG AS CENTRAL OFFICE US NAVAL AIR SYS CMD WASHINGTON DC 20361
1	CDR ATTN CODE AIR 516J LTC EXUM US NAVAL AIR SYS CMD WASHINGTON DC 20361
1	CDR ATTN CODE AIR 41112I T FURLOUGH ADR PROGRAM MANAGER US NAVAL AIR SYS CMD WASHINGTON DC 20361-4110
2	CDR ATTN D H HALL CODE 3181 T HORTON CODE 3386 US NAVAL WEAPONS CTR CHINA LAKE CA 93555-6001
1	NAVAL POSTGRADUATE SCHOOL ATTN PROF R E BALL DEPT OF AERONAUTICS & ASTRONAUTICS MONTEREY CA 93943
1	CDR ATTN P WEINBERG JTCG AS5 AIR-516J5 NAVAL AIR SYS CMD WASHINGTON DC 20361-5160

NO. OF COPIES	ORGANIZATION
1	COMMANDER ATTN CODE GZ4 T WASMUND 17320 DAHLGREN RD US NAVAL SURFACE WARFARE CENTER DAHLGREN VA 22448-5100
1	COMMANDER ATTN CODE PO3912 B NOFREY USA NAVAL AIR WEAPONS CENTER CHINA LAKE CA 93555-6001
1	CDR ATTN CODE 1210 S N GOLDSTEIN CARDEROCK DIVISION NAVAL SURFACE WARFARE CENTER BETHESDA MD 20084-5000
1	CDR ATTN CPT OCONNELL GSM ALC TIED MCCLELLAN AFB CA 95652
2	CDR ATTN G BENNETT M LENTZ ASB XRM WRIGHT PAT AFB OH 45433
1	CDR ATTN K NELSON FTD SDMBU WRIGHT PAT AFB OH 45433
1	CDR ATTN R REINHARDT FTD WRIGHT PAT AFB OH 45433
1	CDR ATTN J SUGRUE FTD SDAEA WRIGHT PAT AFB OH 45433
1	CDR ATTN V VELTON AFWAL AARA WRIGHT PAT AFB OH 45433
1	CDR ATTN D VOYLS WL FIVS WRIGHT PAT AFB OH 45433

NO. OF
COPIES ORGANIZATION

1 CDR
ATTN AFATL DLY
J B FLINT
AIR FORCE ARMAMENT LABORATORY
EGLIN AFB FL 32542-5000

2 CMDR
ATTN CDJ
CPT JOST
J FAISON
AIR FORCE WRIGHT AERO LABS
WRIGHT PAT AFB OH 45433-6523

1 CDR
ATTN ATZQ CDC
HQ USA AVIATION CTR
FT RUCKER AL 36362-5000

1 CDR
ATTN ATZQ TDS SM
HQ USA AVIATION CTR
FT RUCKER AL 36352-5000

1 DIR
8120 WOODMONT AVE
USA CONCEPTS ANALYSIS AGENCY
BETHESDA MD 20814-2797

1 BOARD ON ARMY SCI & TECHNOLOGY
NATIONAL RESEARCH COUNCIL
ROOM MH 280
2101 CONSTITUTION AVE NW
WASHINGTON DC 20418

1 INSTITUTE FOR DEFENSE ANALYSIS
ATTN MR I A KAUFMAN
1801 N BEAUREGARD ST
ALEXANDRIA VA 22311

1 INSTITUTE FOR DEFENSE ANALYSIS
ATTN C F KOSSACK
1005 ATHENS WAY
SUN CITY FL 33570

1 INSTITUTE FOR DEFENSE ANALYSIS
ATTN DR N SUBRAMONIAN
14309 HOLLYHOCK WAY
BURTONSVILLE MD 20866

NO. OF
COPIES ORGANIZATION

1 ADPA
ATTN D R ALEXANDER
TWO COLONIAL PL STE 400
2101 WILSON BLVD
ARLINGTON VA 22201-3061

1 AFELM THE RAND CORPORATION
ATTN LIBRARY D
1700 MAIN ST
SANTA MONICA CA 90406

1 ARMORED VEHICLE TECHNOLOGIES
ATTN C M EDWARDS
PO BOX 2057
WARREN MI 48090

1 BATTELLE
ATTN R GOLLY
EDGEWOOD OPERATIONS
2113 EMMORTON PK RD
EDGEWOOD MD 21040

1 THE BDM CORPORATION
ATTN E J DORCHAK
7915 JONES BRANCH DR
MCLEAN VA 22102-3396

1 BELL HELICOPTER TEXTRON
ATTN J R JOHNSON
PO BOX 482
FT WORTH TX 76101

3 BMY
ATTN W J WAGNER JR
R W JENKINS
E MAGALSKI
PO BOX 1512
YORK PA 17404

1 DENVER RESEARCH INSTITUTE
ATTN L G ULLYATT
BW 228
2050 E ILIFF AVE
DENVER CO 80208

1 FMC CORPORATION
ATTN S KRAUS
1105 COLEMAN AVE BOX 1201
SAN JOSE CA 95108

NO. OF COPIES	ORGANIZATION
1	FMC CORPORATION ATTN R S BECK 881 MARTIN AVE SANTA CLARA CA 95052
1	KAMAN SCIENCES CORPORATION ATTN T S PENDERGRASS 600 BOULEVARD S STE 208 HUNTSVILLE AL 35802
1	LOGISTICS MANAGEMENT INSTITUTE ATTN J WALLICK 6400 GOLDSBORO RD BETHESDA MD 20817-5886
1	OKLAHOMA STATE UNIVERSITY ATTN T M BROWDER JR COLLEGE OF ENG ARCHITECTURE & TECH PO BOX 1925 EGLIN AFB FL 32542
1	TASC ATTN R E KINSLER 1992 LEWIS TURNER BLVD FT WALTON BEACH FL 32548-1255
1	SURVICE ENGINEERING ATTN J FOULK 1003 OLD PHILADELPHIA RD STE 103 ABERDEEN MD 21001
	<u>ABERDEEN PROVING GROUND, MD</u>
5	DIR, USAMSAA ATTN: AMXSY-E, W. CLIFFORD AMXSY-C, J. KRAMAR AMSXY-J, A. LAGRANGE AMSXY-L, M. K. STEINER AMSXY-D, J. MCCARTHY
5	CDR, USATECOM ATTN: AMSTE-CG AMSTE-TA-L, N. HARRINGTON
6	CDR, USAOC&S ATTN: ATSL-DES ATSL-DTD-CM ATSL-CD-TE ATSL-CD-BDAR (HOY) (3 CP)

NO. OF COPIES	ORGANIZATION
40	DIR, USARL ATTN: AMSRL-SL-B, P. DEITZ (4 CP)(328) W. WINNER (328) AMSRL-SL-BA, J. WALBERT (1065) R. BOWERS (1065) M. BURDESHAW (1065) S. POLYAK (1065) L. ROACH (3 CP) (1065) R. WALTHER (1065) W. WARFIELD (1065) E. WEAVER (1065) AMSRL-SL-BG, A. YOUNG (238) W. HURZ (238) L. LOSIE (238) J. PLOSKONKA (247) J. ROBERTSON (238) L. WILSON (238) AMSRL-SL-BL, M. RITONDO (328) E. DAVISSON (328) J. HUNT (328) S. JUARASCIO (328) D. LYNCH (328) AMSRL-SL-BS, D. BELY (328) R. GROTE (328) AMSRL-SL-BV, J. MORRISSEY (247) K. APPLIN (247) W. MERMAGEN JR. (4 CP)(247) G. MOSS (247) K. MURRAY (247) AMSRL-SL-I, M. STARKS (433) D. HASKELL (433) S. HENRY (433) M. VOGEL (433)

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number ARL-TR-738 Date of Report April 1995

2. Date Report Received ____________________

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ____________________

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) ____________________

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. ____________________

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) ____________________

CURRENT ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

Organization

Name

Street or P.O. Box No.

City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)

(DO NOT STAPLE)

DEPARTMENT OF THE ARMY

OFFICIAL BUSINESS

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 0001,APG,MD

POSTAGE WILL BE PAID BY ADDRESSEE

DIRECTOR
U.S. ARMY RESEARCH LABORATORY
ATTN: AMSRL-SL-BA
ABERDEEN PROVING GROUND, MD 21005-5068